

AN - 93-127927
TI - TASK TERMINATION PROCESSOR
PA - (2000522) FUJITSU LTD
IN - KATAI, YOSHIYUKI
PN - 93.05.25 J05127927, JP 05-127927
AP - 91.11.05 91JP-288196, 03-288196
SO - 93.09.09 SECT. P, SECTION NO. 1610; VOL. 17, NO. 502, PG. 99.
IC - G06F-009/46
JC - 45.1 (INFORMATION PROCESSING--Arithmetic Sequence Units)
AB - PURPOSE: To obtain a task termination processor which can comparatively and easily realize a function equal to a task termination exit on the release processing of a computer resource, which is remained when a task terminates.

CONSTITUTION: A task generation part 3 requests to generate a termination processing part 4 as a slave task where a relevant task 5 is set to be a slave task to an operating system 1 by the prescribed request of the task. The termination processing part 4 has a prescribed termination processing routine 6, designates the termination processing routine, requests the registration of an abnormal termination exit to the operating system 1, designates a prescribed event, sets a system to an event waiting state. When the master task 5 terminates, the operating system 1 judges the system to be in an abnormal termination state when the task of the termination processing part 4 is not terminated, and starts the termination processing routine 6 in accordance with the registration of the abnormal termination exit. The termination processing routine 6 executes the processing for releasing the computer resource concerned as to the computer resource which the master task 5 uses.

(51)Int.Cl.⁵

G 0 6 F 9/46

識別記号

3 4 0 B 8120-5B

庁内整理番号

F 8120-5B

F I

技術表示箇所

審査請求 未請求 請求項の数1(全 5 頁)

(21)出願番号

特願平3-288196

(22)出願日

平成3年(1991)11月5日

(71)出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中1015番地

(72)発明者 片井 喜之

静岡県静岡市伝馬町16番地3 株式会社富

士通静岡エンジニアリング株式会社内

(74)代理人 弁理士 井桁 貞一

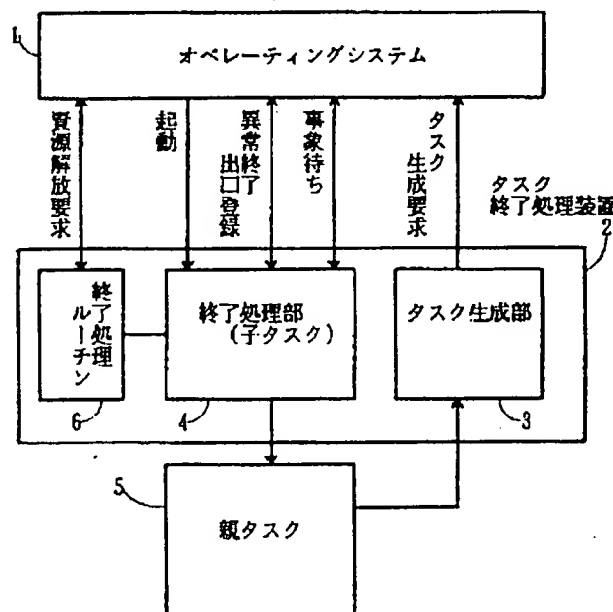
(54)【発明の名称】 タスク終了処理装置

(57)【要約】

【目的】 タスク終了時に残された計算機資源の解放処理に関し、タスク終了出口と同等の機能を、比較的簡単に実現できるようにするタスク終了処理装置を目的とする。

【構成】 タスク生成部3は、該タスクの所定の要求によって、終了処理部4を当該タスクを親タスク5とする子タスクとして生成することをオペレーティングシステム1に要求し、終了処理部4は、所定の終了処理ルーチン6を有し、終了処理ルーチン6を指定して、該異常終了出口の登録をオペレーティングシステム1に要求した後、所定の事象を指定して事象待ち状態となり、オペレーティングシステム1は、親タスク5の終了時に、終了処理部4のタスクが終了していなければ、該異常終了状態と判断して、該異常終了出口の登録に従って終了処理ルーチン6を起動し、終了処理ルーチン6は、親タスク5の使用していた計算機資源について、当該計算機資源を解放する処理を実行するように構成する。

本発明の構成を示すブロック図



【特許請求の範囲】

【請求項1】 計算機の実オペレーティングシステム(1)は、タスクの要求を受けて異常終了出口を登録することにより、当該タスクが所定の異常終了状態の場合に該登録に指定されたルーチンを起動する機能を有し、該オペレーティングシステム(1)の制御下にある装置(2)であって、タスク生成部(3)と終了処理部(4)とを有し、該タスク生成部(3)は、該タスクの所定の要求によって、該終了処理部(4)を当該タスクを親タスク(5)とする子タスクとして生成することを該オペレーティングシステム(1)に要求し、
10 該終了処理部(4)は、所定の終了処理ルーチン(6)を有し、該終了処理ルーチンを指定して、該異常終了出口の登録を該オペレーティングシステム(1)に要求した後、所定の事象を指定して事象待ち状態となり、
該オペレーティングシステム(1)は、該親タスク(5)の終了時に、該終了処理部(4)のタスクが終了していなければ、該異常終了状態と判断して、該異常終了出口の登録に従って該終了処理ルーチン(6)を起動し、
20 該終了処理ルーチン(6)は、該親タスク(5)の使用していた計算機資源について、当該計算機資源を解放する処理を実行するように構成されていることを特徴とするタスク終了処理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、計算機において、タスク終了時に残された計算機資源の、解放処理を行うためのタスク終了処理装置に関する。

【0002】

【従来の技術と発明が解決しようとする課題】 公知のように計算機で実行されるタスクが終了する場合には、使用していた共用の計算機資源、即ち主記憶領域や共用データセット等をシステムに返却する手続きを行って、それらを他のタスクで使用できる状態に解放する処理を行った後に終了するのが原則である。

【0003】 しかし、そのような共用資源の解放処理を行うことなくタスクを終了する場合があります、その状態を放置すると、使用できる共用資源が次第に減少し、遂にはシステムが正常に稼働できない状態になり得るので、そのようにタスク終了時に残されている諸資源の解放処理を行うために、オペレーティングシステムでタスク終了出口のサービスが設けられている。

【0004】 このタスク終了出口を利用するように登録されていると、登録されているタスクが終了する時には、オペレーティングシステムのタスク終了処理の中でタスク終了出口ルーチンが起動されて、必要な資源の解放処理を行うので、共用資源が未解放のままに放置されることが無い。

【0005】 しかし、タスク終了出口の登録は、オペレーティングシステムのシステム生成時に行うことになっ

ているので、その後に追加されるプログラムでは、タスク終了出口のサービスを受けるようにすることが容易にできない。

【0006】 オペレーティングシステムの類似のサービスとして、公知の異常終了出口があり、このサービスではタスクが必要な異常終了出口の登録をオペレーティングシステムに要求して登録することができる。

【0007】 この登録が行われていると、そのタスクが所定の正常状態で終了しない場合に、オペレーティングシステムは、その終了処理の中で、異常終了出口に登録されている異常終了出口ルーチンを起動する。

【0008】 従って、異常終了出口ルーチンによって、例えば異常発生時に必要な後始末等で、そのタスクに固有の処理等を行うようにして、このサービスを有効に利用することができる。

【0009】 しかし、前記のとおり異常終了出口の登録は、タスクが異常終了する場合にしか有効に作用しないので、異常終了出口のサービスを、そのまゝタスク終了出口の代わりに利用することはできない。

【0010】 本発明は、前記のタスク終了出口と同等の機能を、比較的簡単に実現できるようにするタスク終了処理装置を目的とする。

【0011】

【課題を解決するための手段】 図1は、本発明の構成を示すブロック図である。図はタスク終了処理装置の構成であって、計算機の実オペレーティングシステム1は、タスクの要求を受けて異常終了出口を登録することにより、当該タスクが所定の異常終了状態の場合に該登録に指定されたルーチンを起動する機能を有するものとする。

【0012】 タスク終了処理装置2は、オペレーティングシステム1の制御下にある装置であって、タスク生成部3と終了処理部4とを有する。タスク生成部3は、該タスクの所定の要求によって、終了処理部4を当該タスクを親タスク5とする子タスクとして生成することをオペレーティングシステム1に要求する。

【0013】 終了処理部4は、所定の終了処理ルーチン6を有し、終了処理ルーチン6を指定して、該異常終了出口の登録をオペレーティングシステム1に要求した後、所定の事象を指定して事象待ち状態となる。

【0014】 オペレーティングシステム1は、親タスク5の終了時に、終了処理部4のタスクが終了していなければ、該異常終了状態と判断して、該異常終了出口の登録に従って終了処理ルーチン6を起動する。

【0015】 終了処理ルーチン6は、親タスク5の使用していた計算機資源について、当該計算機資源を解放する処理を実行する。

【0016】

【作用】 本発明のタスク終了処理装置により、あるタスクの適当な要求発生を契機として、本装置のタスク生成

部がオペレーティングシステムによって子タスクを生成する。

【0017】この子タスクは、終了処理部として機能し、オペレーティングシステムの異常終了登録サービスを使用して、終了処理ルーチンを登録した後、所定の事象を指示して事象待ち状態になる。

【0018】親タスクが、本来の所要の処理を終了して、タスクを終了しようとし、オペレーティングシステムの終了処理に入ると、オペレーティングシステムは通常の制御として、子タスクである終了処理部が事象待ち状態のまゝで、終了していないことを検出する。

【0019】従ってオペレーティングシステムは、親タスクが終了する時点で子タスクが生きていることにより、異常状態と判断して、子タスクを強制的に異常終了させることにし、異常終了出口の登録があるために、そこに指定されている終了処理ルーチンを起動する。

【0020】そこで、終了処理ルーチンを、例えば前記タスク終了出口ルーチンに代わって同等の終了処理を行うように、処理内容を定義しておくことにすれば、タスク終了出口に代わるサービスを比較的容易に実現することができる。

【0021】

【実施例】計算機システムが、例えば図形処理等を実行するための各種関数を提供する基本プログラムがあり、図形処理の応用プログラムは、それらの関数を実行するための関数呼出の列で構成するようになっているとする。

【0022】そのようなシステムにおいては、例えば応用プログラムが、最初に図形処理に必要な諸計算機資源の初期化を、初期化関数を呼び出すことによって行うので、その初期化関数の初期化処理の一部として、本発明のタスク終了処理装置を組み込めば、応用プログラムでは陽にタスク終了処理装置への要求送出を意識しなくてもよい。

【0023】図2は、そのような基本プログラムを使用して処理を実行する応用プログラムを有するシステムにおける、本発明の実施例の処理の流れを示す図であり、応用プログラムのタスクが起動すると先ず処理ステップ10で基本プログラムの初期化関数を呼ぶ。

【0024】初期化関数は処理ステップ11で図1のタスク生成部3の機能を実行して、子タスクの生成要求を出し、オペレーティングシステムによって処理ステップ12により、子タスクが起動される。

【0025】子タスクは図1の終了処理部4を構成し、処理ステップ13で異常終了出口の登録を要求し、オペレーティングシステムにより処理ステップ14で登録が行われ、終了処理ルーチンが異常終了出口ルーチンとして登録される。

【0026】子タスクは、処理ステップ15で初期化関数に異常終了出口の登録済を通知した後、処理ステップ16

で所定の事象を指定して事象待ちをオペレーティングシステムに要求し、処理ステップ17の処理で事象待ち状態となる。

【0027】初期化関数では、子タスクから登録済の通知を受けると、処理ステップ18で、本来の初期化処理を開始して資源獲得要求を出すので、オペレーティングシステムは処理ステップ19で資源割当を行う。

【0028】資源の割当が終わると、初期化関数は処理ステップ20で処理を終わり応用プログラムに制御を戻すので、処理ステップ21で応用プログラムによる業務自体の処理が実行される。

【0029】応用プログラムの処理が終わると処理ステップ22でオペレーティングシステムへ制御が渡り、処理ステップ23でタスクの終了処理が行われ、その結果子タスクが事象待ち状態のまゝであるので、処理ステップ24で子タスクの強制終了処理を始める。

【0030】その結果処理ステップ25で、異常終了出口に登録されている終了処理ルーチンが起動され、処理ステップ26で、応用プログラムのタスクが獲得した資源について、オペレーティングシステムへ資源の返却が行われ、処理ステップ27でオペレーティングシステムにより資源が解放される。

【0031】なお、基本プログラムに例えば終了化関数として、資源の返却等を行う関数が準備されている場合には、終了化関数の処理の一部として、例えば前記で子タスクが待つ事象を発生させることにより、子タスクを実行させる。

【0032】そこで子タスクは自身で正常に終了するようにすれば、応用プログラムが終了化関数を呼んで終了する場合には、子タスクの登録した異常終了出口による終了処理ルーチンの起動が行われなくすることができる。

【0033】

【発明の効果】以上の説明から明らかなように本発明によれば、計算機における、タスク終了時に残された計算機資源の解放処理のために、タスク終了出口と同等の機能を、比較的簡単に実現できるという著しい工業的效果がある。

【図面の簡単な説明】

【図1】 本発明の構成を示すブロック図

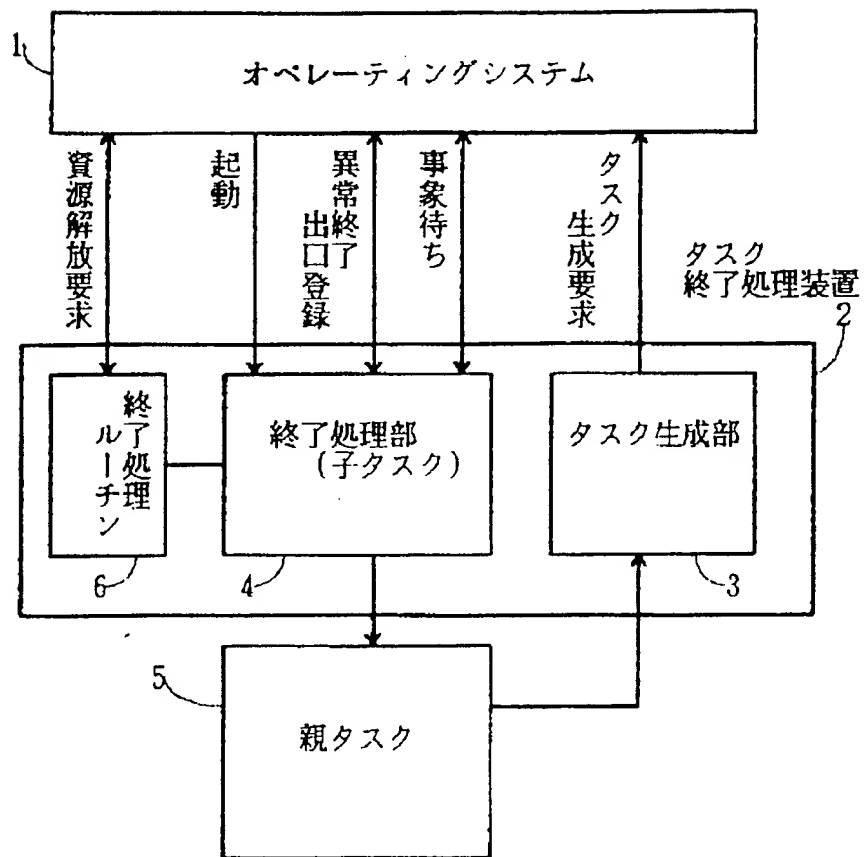
【図2】 本発明の処理の流れ図

【符号の説明】

- 1 オペレーティングシステム
- 2 タスク終了処理装置
- 3 タスク生成部
- 4 終了処理部
- 5 親タスク
- 6 終了処理ルーチン
- 10~27 処理ステップ

【図1】

本発明の構成を示すブロック図



【図2】

本発明の処理の流れ図

